EUROPA Quick Start

This tutorial provides just enough information for you to create and run simple planning problems using EUROPA. Once you're familiar with the basics, see the Documentation page for more details.

EUROPA can be used to solve Constraint Programming, Scheduling and Planning problem. For the purposes of this tutorial we will use a simple planning problem.

These are the main steps that a EUROPA user will normally follow to solve a problem :

- Create a model of the problem domain using NDDL, EUROPA's modeling language
- Create a particular instance of the problem using NDDL
- Instantiate a solver and ask it to look for a solution
- Inspect the plan, possibly modify the problem and re-plan
- When the model is working as expected, probably embed it into a larger application

Now, for our example, let's assume that we will create a plan to turn a light bulb on and off.

For this this simple domain here is what we would do (The files for this example can be found under PlanWorks/PSUI/test/Light).

TODO: User must have build EUROPA already and have PLASMA and PlanWorks checked out

Create a NDDL model for the domain

The New Domain Description Language (NDDL) is a simple but powerful language used to describe both problem domains and problem instances in EUROPA

```
// Light-model.nddl

#include "Plasma.nddl"

class LightSwitch extends Timeline
{
    predicate turnOn  { duration=1; }
    predicate turnOff { duration=1; }
}

class LightBulb extends Timeline
{
    LightSwitch mySwitch_;

    LightBulb(LightSwitch b)
    {
```

```
        mySwitch_ = b;
    }

    predicate On  {}
    predicate Off {}
}

LightBulb::On
{
    met_by(object.Off);              // Bulb must be Off to be turned On
    met_by(object.mySwitch_.turnOn); // Must be turned on through the switch
}

LightBulb::Off
{
    met_by(object.On);                // Bulb must be On to be turned Off
    met_by(object.mySwitch_.turnOff); // Must be turned off through the switch
}
```

TODO: describe this model line by line

Create a problem instance

```
// Light-initial-state.nddl

#include "Light-model.nddl"

// Problem instance : turning the light off
LightSwitch switch1 = new LightSwitch();
LightBulb bulb1 = new LightBulb(switch1);

// At time 0, the bulb is on
fact(bulb1.On initialCondition);
eq(initialCondition.start,0);

// We want the bulb to be off by time 10
goal(bulb1.Off goal1);
lt(0,goal1.start);
lt(goal1.start,10);
```

TODO: describe this line by line

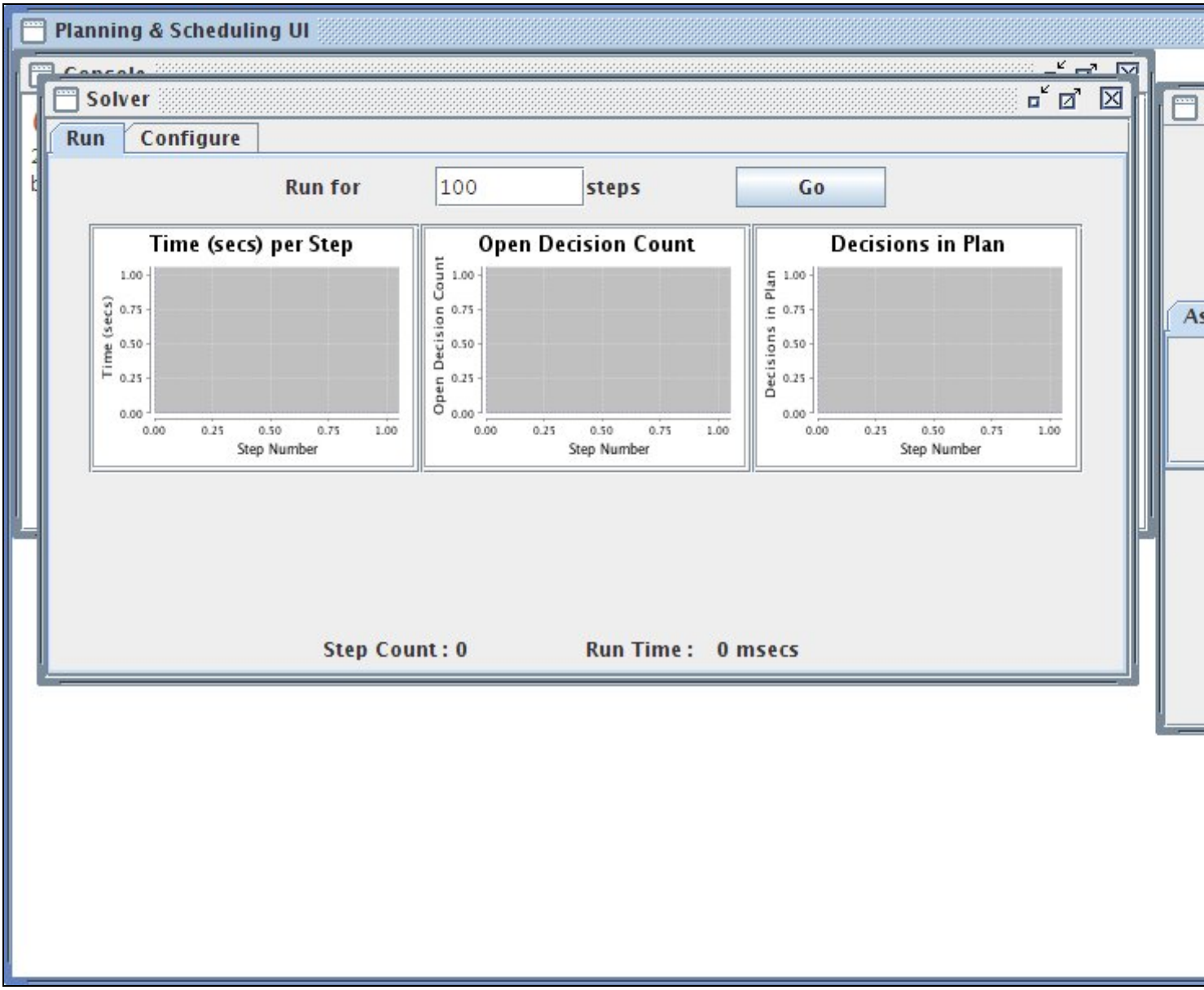
Instantiate and run a Solver

Run PSUI : (TODO, installation must be complete, environment set up)

```
% cd PlanWorks/PSUI/test
% ant -Dproject=Light
Buildfile: build.xml

init:

compile:

run:
```

```
[echo] Running Light project
[java]  autoWrite 0
[java] [XMLInterpreter:InterpretedObject]Initialized variable:bulb1.mySwitch_ to OBJECT-LightSwitc
```

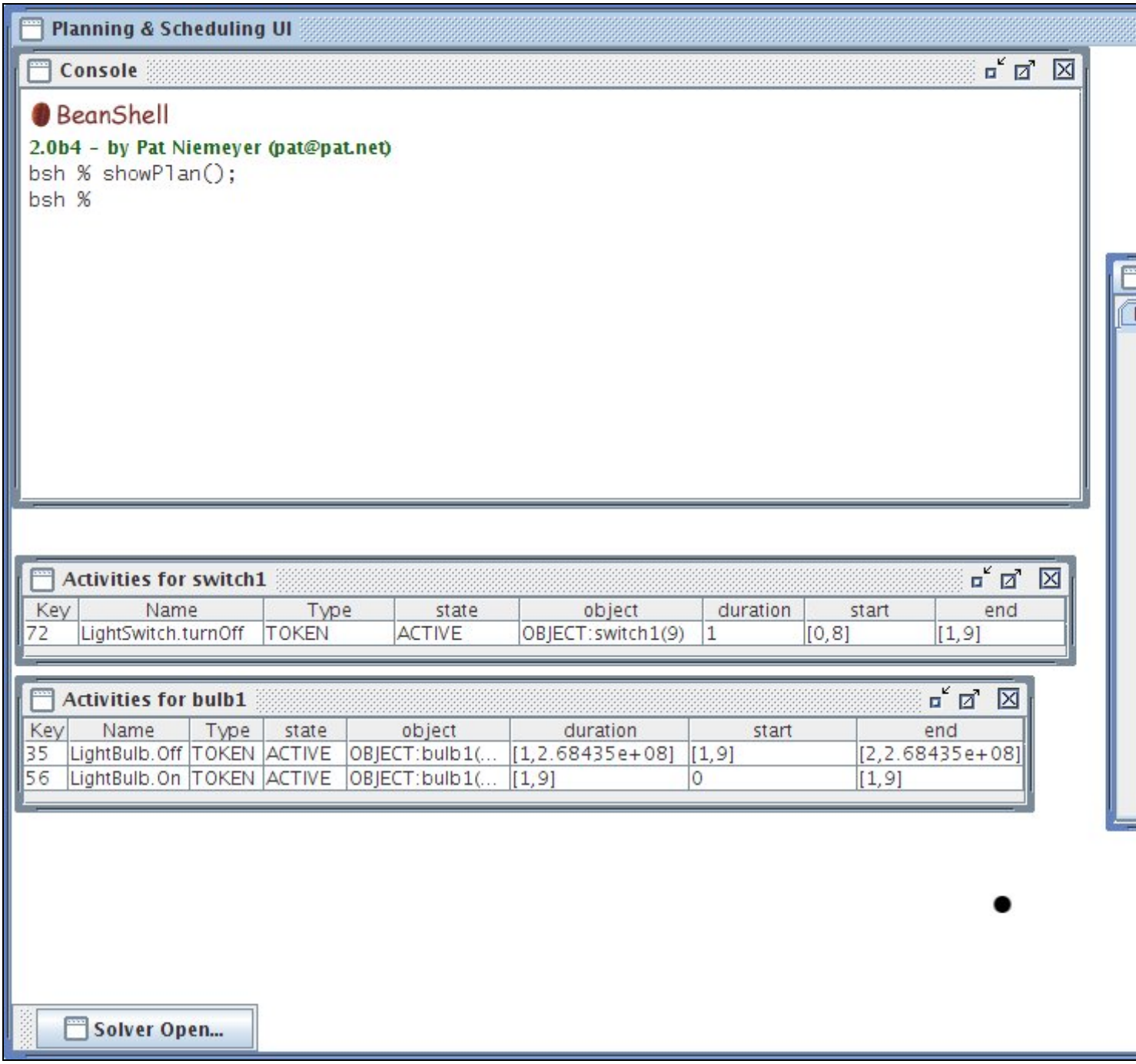
You should see the following window come up :



In the "Solver" window, hit the "Go" button to have the solver run. The solver should finish after 7 steps, satisfying the goal that was specified in Light-initial-state.nddl (that is, to turn the light off by time=10).

Examine Results

You can now run the method showPlan() defined in Light.bsh?



You can now see that the planner decided that : bulb1 is On from [] to [] and Off from [] to [] lightSwitch1 is turned off at time []

The intervals mean that the token can start/end at any point in the specified interval.

Embed your model into an application

Sometimes what we have done so far is all you may want to do : create a model, a problem instance, generate a plan, visualize the results, and perhaps dump them to a file.
You may also want to embed this model as part of a bigger application, in that case, you will need to link in EUROPA as a library into your application and use the programmatic API (Java or C++) to perform the steps described above.
the details of how to embed EUROPA into your application are provided here

TODO: describe briefly how to embed into a Java application, provide link for detail of how to embed in Java and C++

Examples

Now that we have cover the entire cycle with a simple example, see the Examples page to see more advanced examples that can serve as starting points for your own model, or just to learn more about what EUROPA can do.

Next Steps

TODO: provide links

- How to create your own project in PSUI
- How to embed your model into a C++ or Java application
- All the details about modeling in NDDL, configuring the built-in solver and using EUROPA's debugging tools.
- Extending EUROPA adding your own constraints, solvers, etc